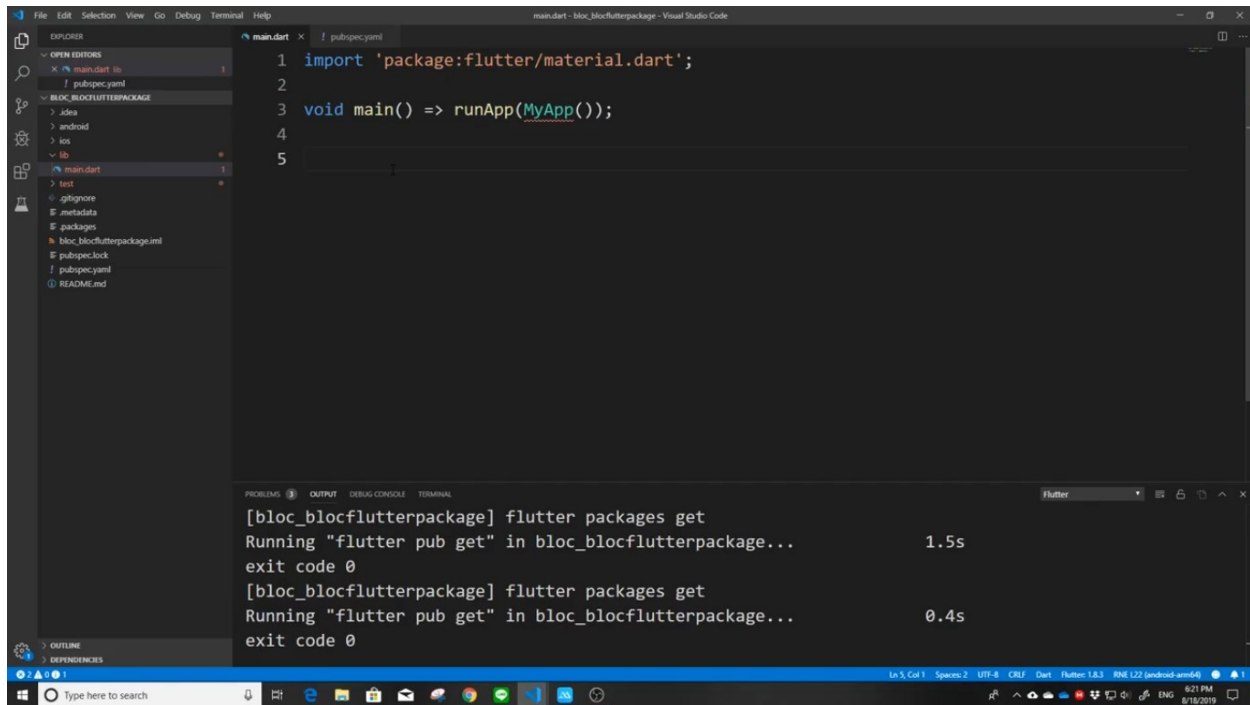


Halo semuanya, selamat malam. Dalam tutorial video ke-44 saya, saya sudah menjelaskan cara menggunakan "BLoC State Management" meskipun dalam tutorial itu, saya menjelaskan penggunaannya menggunakan "Stream Controller" dan "Stream Builder", tanpa menggunakan paket apa pun dari luar

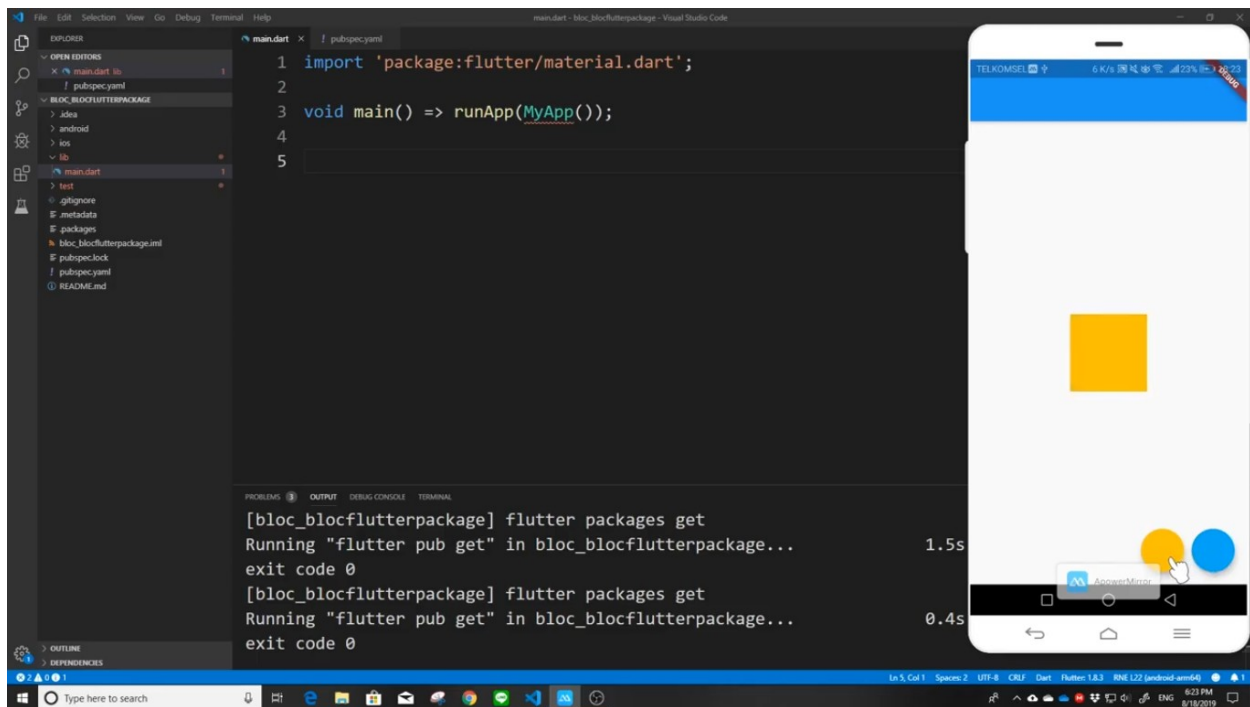


```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5
```

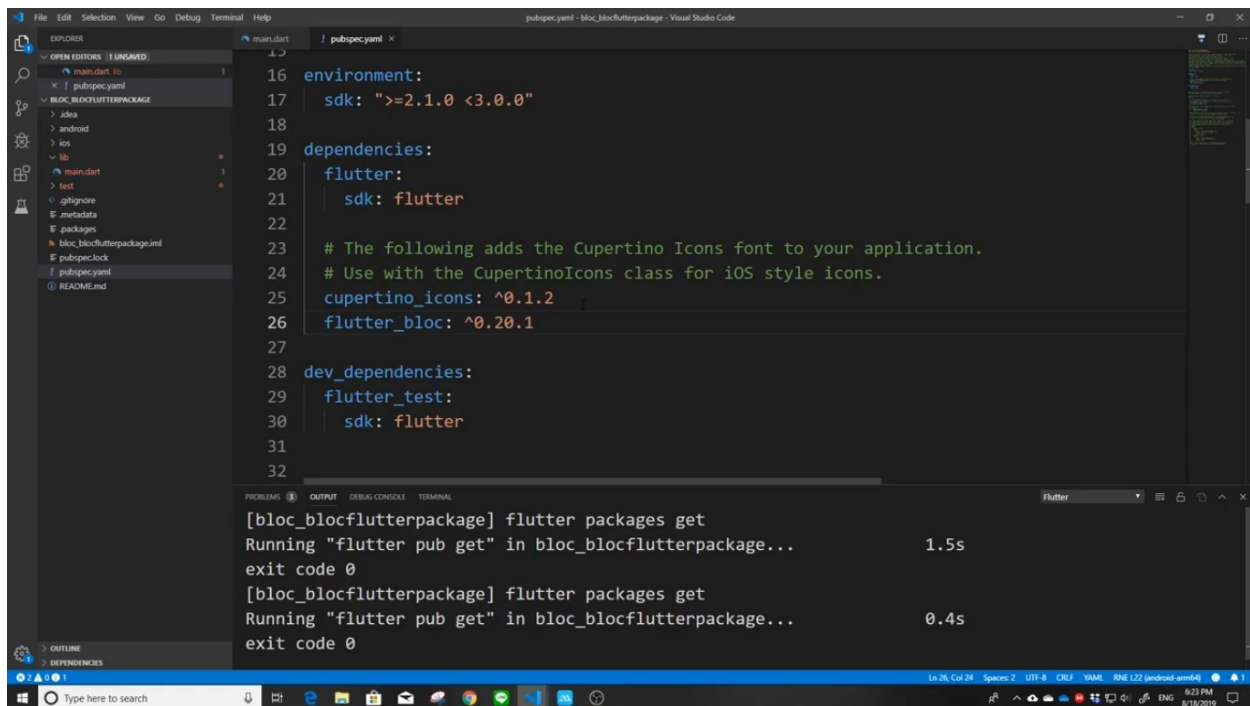
Flutter

```
[bloc_blocflutterpackage] flutter packages get
Running "flutter pub get" in bloc_blocflutterpackage... 1.5s
exit code 0
[bloc_blocflutterpackage] flutter packages get
Running "flutter pub get" in bloc_blocflutterpackage... 0.4s
exit code 0
```

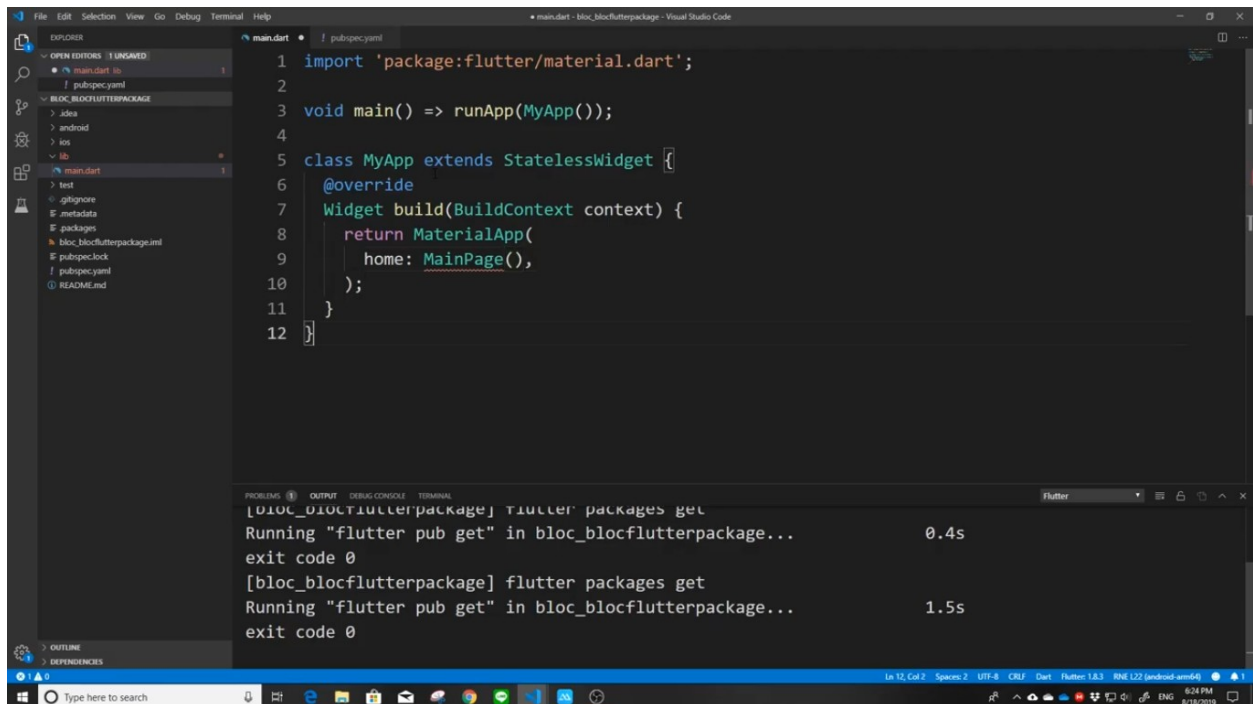
. Nah, dalam tutorial ini saya ingin menunjukkan cara membuat "BLoC State Management" menggunakan paket dari luar. Paket yang akan kita gunakan disebut "Flutter BLoC". Contoh aplikasi yang akan kita buat sama dengan yang kita buat pada tutorial sebelumnya. Jadi, tampilannya akan seperti ini.



Akan ada 'Wadah Animasi', dan dua 'Tombol Melayang' yang jika kita klik, akan berubah warna. OKE. Jadi, sama seperti yang ada di tutorial ke-44 saya. Mari kita mulai. Buka 'pubspec.yaml' terlebih dahulu. Di sini, kita harus menambahkan 'flutter\_bloc' dan kita akan menggunakan versi 0.20.1. Menyimpan.



Sekarang, mari kita ke 'main' dan seperti biasa kita akan membuat widget 'Stateless' beri nama "MyApp", di sini... 'MaterialApp', 'home'. Saya akan memisahkan home dan akan membuat mainpage untuknya secara terpisah.



The screenshot shows the Visual Studio Code editor with the file explorer on the left, the main editor in the center, and the terminal at the bottom. The file explorer shows the project structure with files like main.dart, pubspec.yaml, and various package folders. The main editor displays the code for main.dart, which imports the Flutter material package, defines a main function, and creates a StatelessWidget named MyApp. The MyApp class overrides the build method to return a MaterialApp widget with a home property set to MainPage(). The terminal at the bottom shows the output of the flutter pub get command, indicating that the packages were successfully resolved.

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatelessWidget {
6   @override
7   Widget build(BuildContext context) {
8     return MaterialApp(
9       home: MainPage(),
10    );
11  }
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Flutter

```
[D:\BLOK_flutterpackage] flutter packages get
Running "flutter pub get" in bloc_blocflutterpackage... 0.4s
exit code 0
[bloc_blocflutterpackage] flutter packages get
Running "flutter pub get" in bloc_blocflutterpackage... 1.5s
exit code 0
```

Di sini, saya akan membuat widget 'Stateless' lainnya dan menamainya "MainPage" dan ini di sini akan mengembalikan 'Scaffold'. 'appBar...' , 'judul' akan menjadi "BLoC menggunakan flutter\_bloc". Kemudian, 'body' akan disetel ke 'Center', dan akan memiliki 'AnimatedContainer' dengan lebar 100 dan tinggi 100. Warnanya akan menjadi 'Colors.amber', lalu durasinya akan disetel hingga 500 milidetik.

```
9      home: MainPage(),
10    ); // MaterialApp
11  }
12 }
13
14 class MainPage extends StatelessWidget {
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       appBar: AppBar(title: Text("Bloc dengan flutter_bloc")),
19       body: Center(
20         child: AnimatedContainer(
21           width: 100,
22           height: 100,
23           color: Colors.amber,
24           duration: Duration(milliseconds: 500),
25         ), // AnimatedContainer
26       ), // Center
27     );
28   }
29 }
```

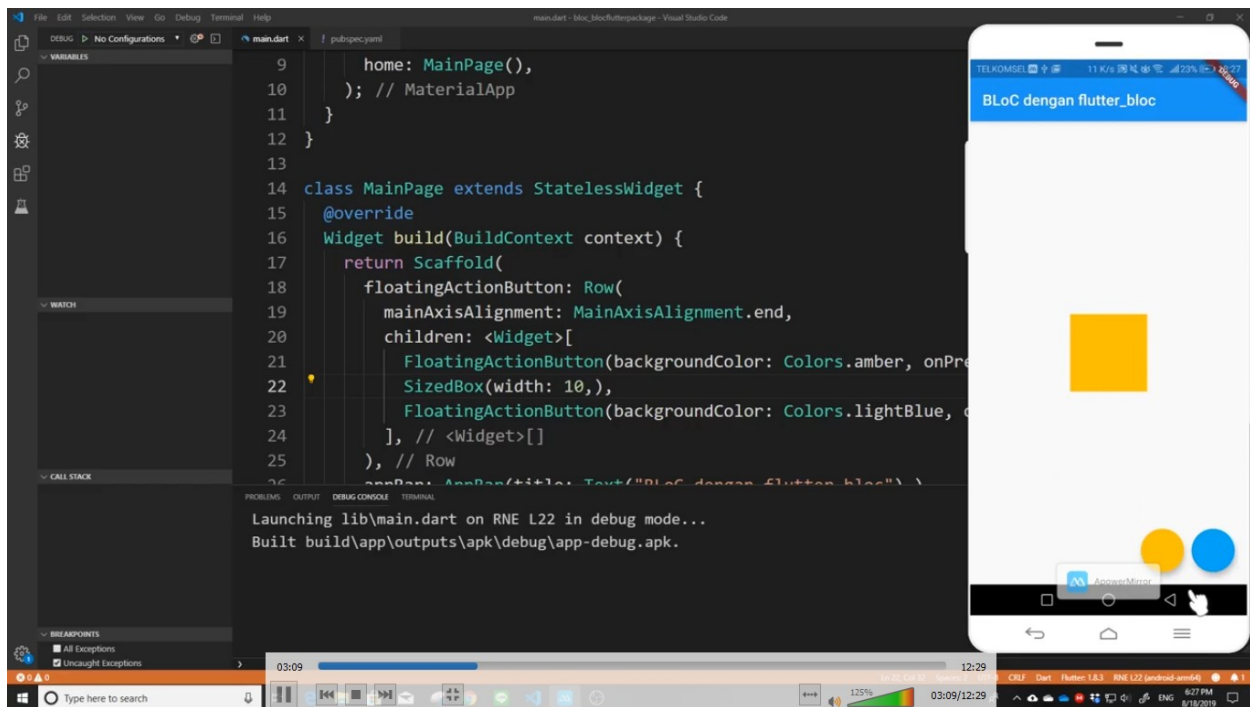
[bloc\_blocflutterpackage] flutter packages get  
Running "flutter pub get" in bloc\_blocflutterpackage... 0.4s  
exit code 0  
[bloc\_blocflutterpackage] flutter packages get  
Running "flutter pub get" in bloc\_blocflutterpackage... 1.5s  
exit code 0

Kemudian di sini, saya akan membuat "floatingActionButton" yang akan ditempatkan di 'Row', 'mainAxisAlignment' nya akan disetel ke 'end' lalu di dalamnya, akan ada dua "FloatingActionButtons". Warnanya akan kuning, untuk tombol pertama, dan 'onPressed' akan dibiarkan kosong untuk saat ini. Salin ini. Untuk yang ini, warnanya biru muda. Di antaranya, saya akan menambahkan beberapa spasi menggunakan 'SizedBox' dengan lebar 10. Seperti itu.

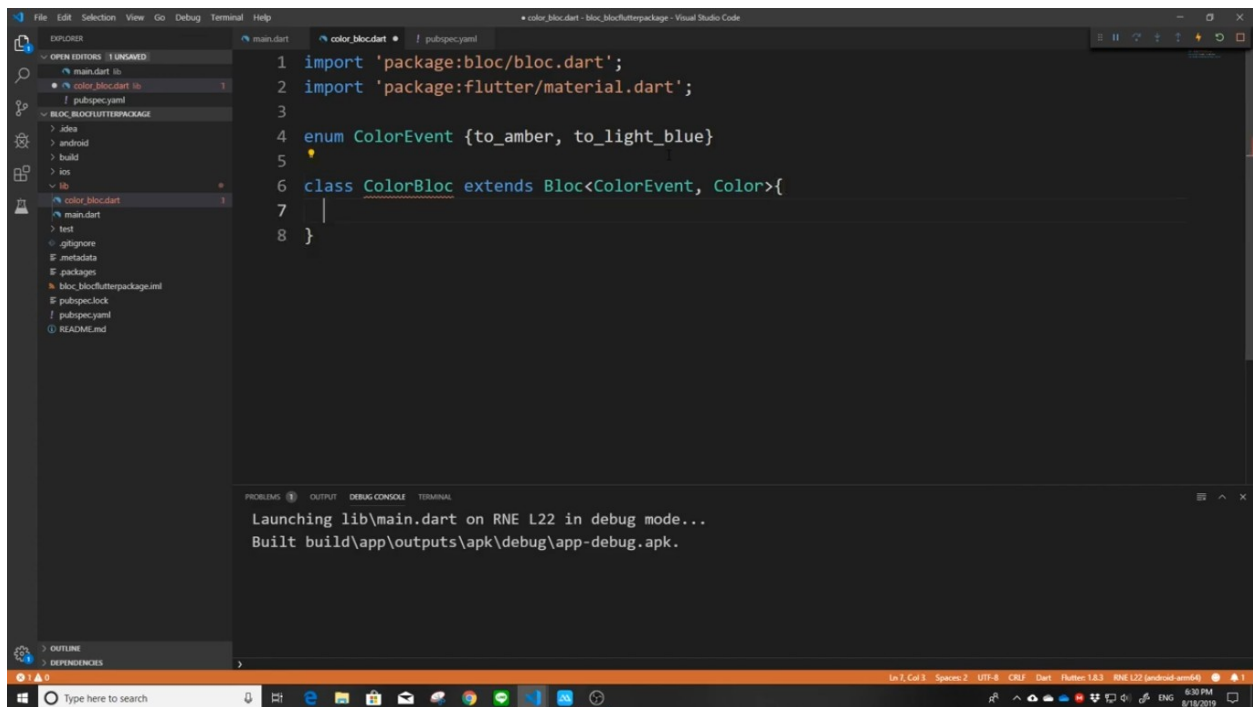
```
9      home: MainPage(),
10    ); // MaterialApp
11  }
12 }
13
14 class MainPage extends StatelessWidget {
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       floatingActionButton: Row(
19         mainAxisAlignment: MainAxisAlignment.end,
20         children: <Widget>[
21           FloatingActionButton(backgroundColor: Colors.amber, onPressed: () {}),
22           SizedBox(width: 10,),
23           FloatingActionButton(backgroundColor: Colors.lightBlue, onPressed: () {}),
24         ], // <Widget>[]
25       ), // Row
26       appBar: AppBar(title: Text("Bloc dengan flutter_bloc")),
27       body: Center(
28         child: AnimatedContainer(
29           width: 100,
30           height: 100,
31           color: Colors.amber,
32           duration: Duration(milliseconds: 500),
33         ), // AnimatedContainer
34       ), // Center
35     );
36   }
37 }
```

[bloc\_blocflutterpackage] flutter packages get  
Running "flutter pub get" in bloc\_blocflutterpackage... 0.4s  
exit code 0  
[bloc\_blocflutterpackage] flutter packages get  
Running "flutter pub get" in bloc\_blocflutterpackage... 1.5s  
exit code 0

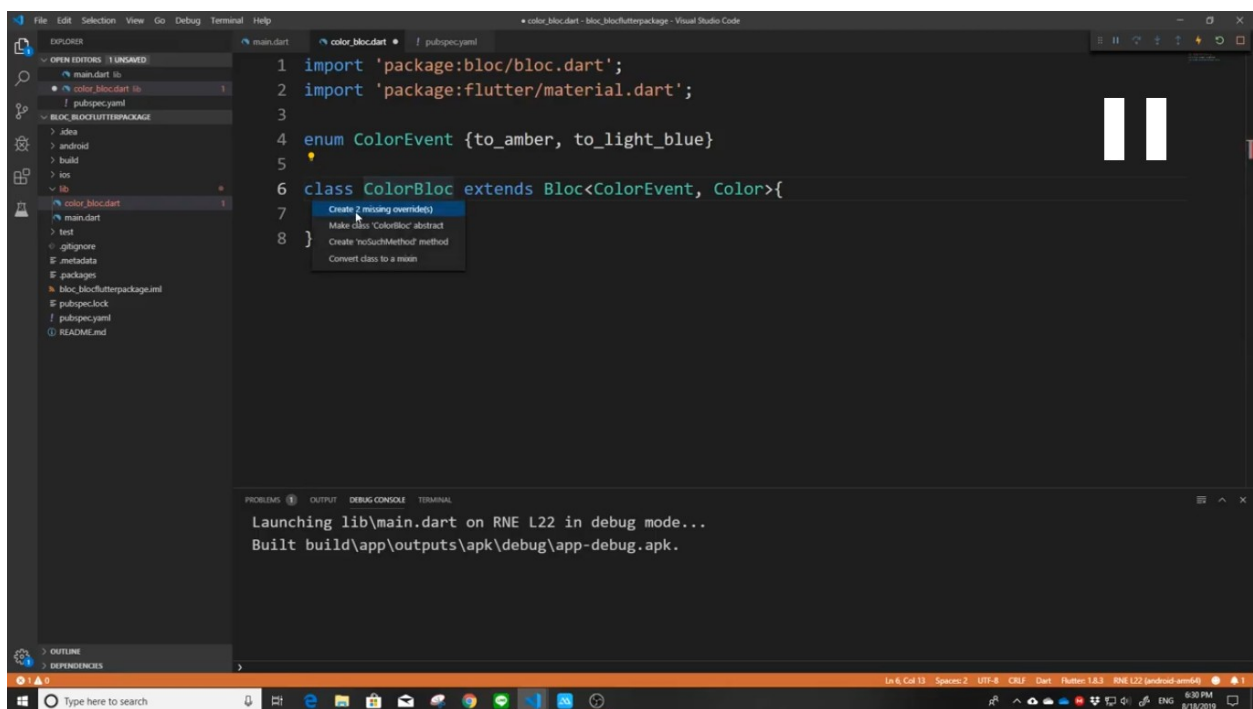
Sekarang simpan, 'F5'. Itu ada. Tata letak utama selesai.



Sekarang, mari kita kembali ke kode. Di sini, kita akan membuat file lain bernama "color\_bloc.dart". Di sini, kita perlu mengimpor "bloc.dart" OK? Serta mengimpor "material.dart". Lalu di sini, kami akan menambahkan 'enum' untuk "ColorEvent". Kalau kamu mulai kurang paham disini seperti misalnya apa yang dimaksud dengan "ColorEvent", kamu bisa cek dulu tutorial saya yang ke 44 (sebelumnya) supaya lebih paham apa yang saya coba jelaskan disini, karena saya tidak akan menjelaskan kembali apa yang sudah saya jelaskan pada tutorial saya sebelumnya. Oke, mari kita lanjutkan. Di sini, akan ada dua jenis, "to\_amber" dan "to\_light\_blue". Selanjutnya, kita akan membuat kelas dan menamainya "ColorBloc" yang akan extends dari 'Bloc'. 'Bloc' ini perlu mengetahui jenis event dan state. Seperti yang kita ketahui, kita mendapatkan event dari "ColorEvent" sementara itu, status yang kita peroleh dari "Color". Jadi di sini, "ColorEvent..." koma, "Color".

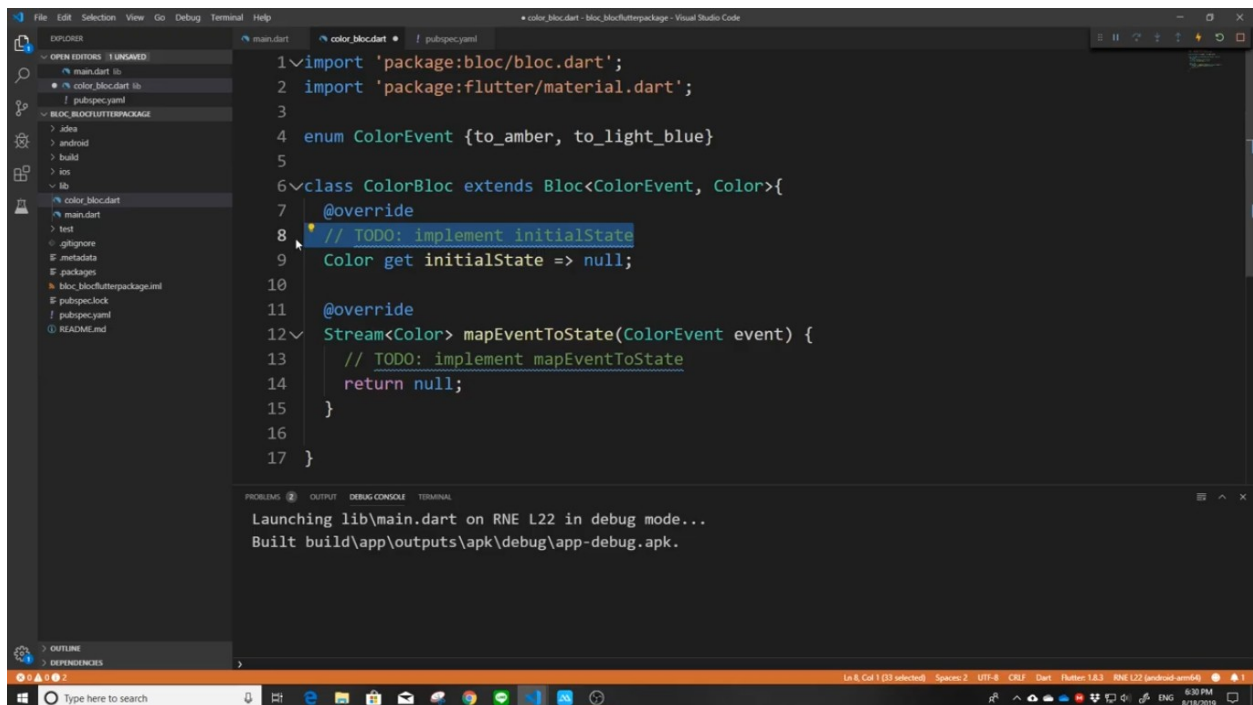


Di sini, kita perlu mengganti dua metode.

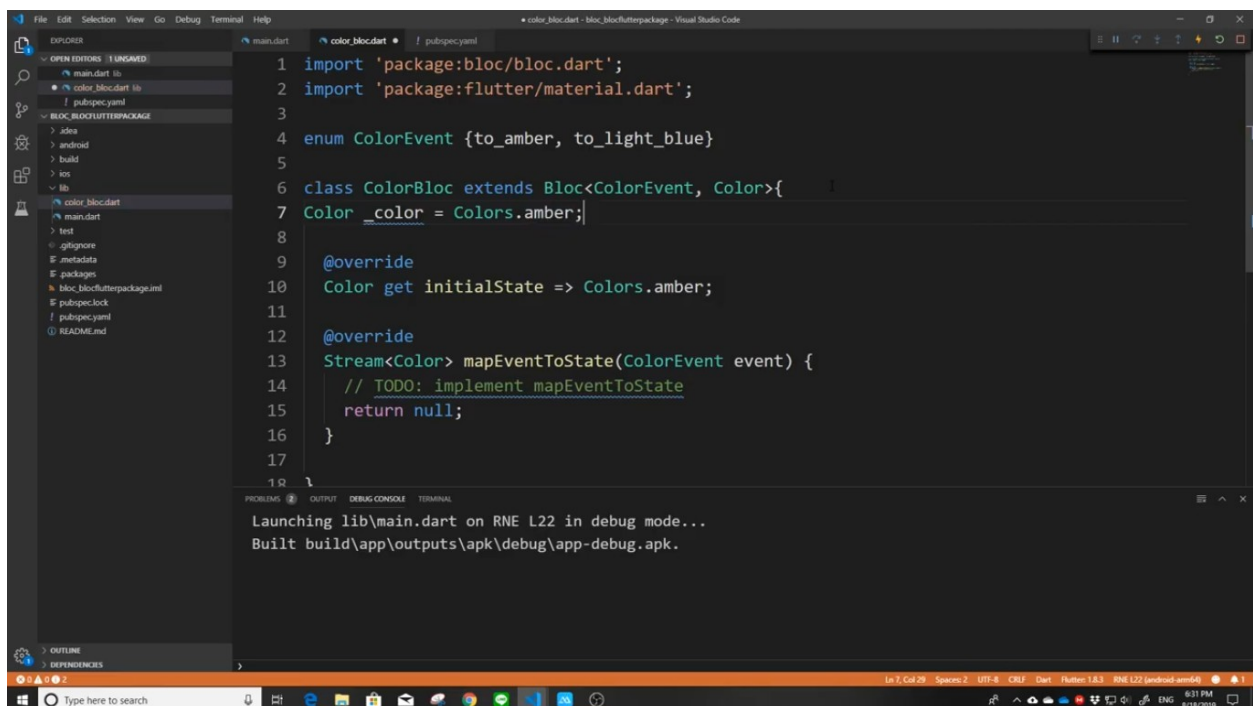


Hapus ini. Yang pertama ditimpa adalah "initalState" dan apa status awal kita?





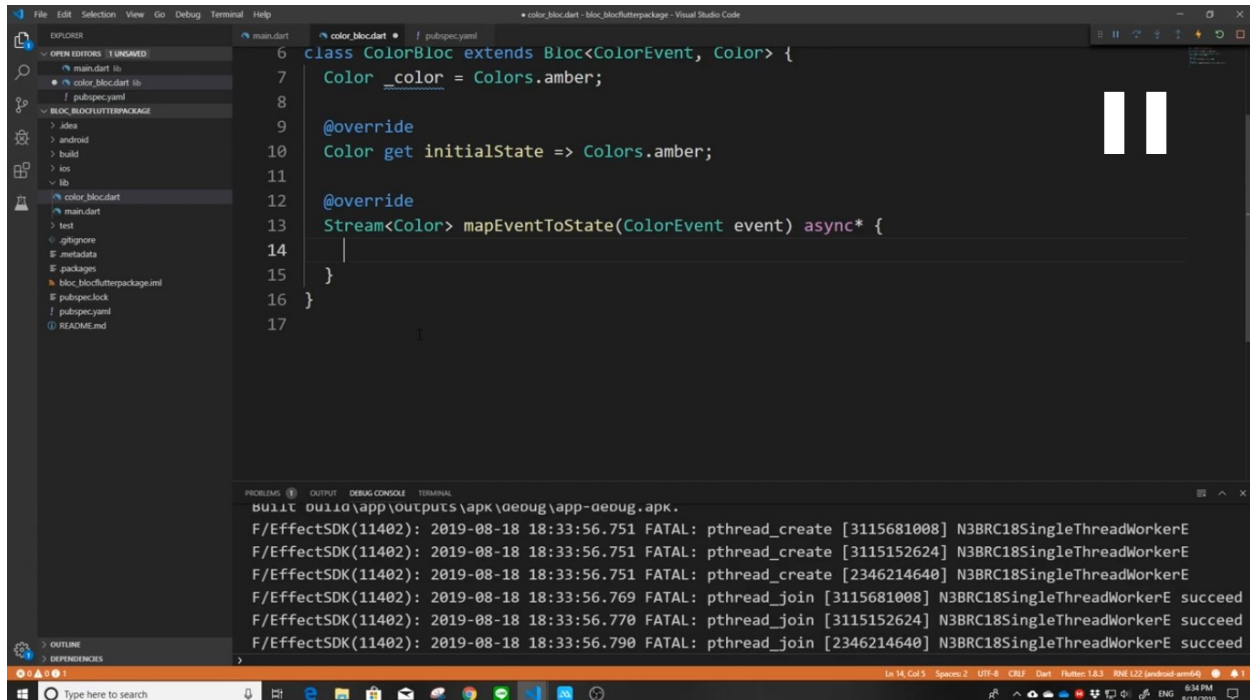
Nah, itu akan menjadi warna pertama yang kita gunakan yaitu amber. Jadi, "initalState" akan berwarna kuning. Kemudian, di dalam ini, kita juga perlu membuat status jadi "Warna..." "\_color" sama dengan titik koma



"Colors.amber" , Sekarang, "Shift-Alt-F".

Kemudian di sini Anda melihat, bahwa ada metode "mapEventToState". Saya sudah menjelaskan bukan? Apa itu "mapEventToState". Pada dasarnya, akan ada peristiwa yang akan dipetakan untuk mengubah keadaan saat ini berdasarkan peristiwa yang

terjadi, kemudian, 'aliran' akan mengalir ke arah 'State Controller' namun, di sini 'State Controller' dan 'Event Controller' sudah terbungkus dalam paket ini - "flutter\_bloc". Oleh karena itu, tidak perlu bingung lagi, yang perlu kita lakukan hanyalah membuat "mapEventToState". "mapEventToState" ini akan mengembalikan 'aliran' karenanya, kami akan mengubah jenisnya menjadi "async\*"



```
6 class ColorBloc extends Bloc<ColorEvent, Color> {
7   Color _color = Colors.amber;
8
9   @override
10  Color get initialState => Colors.amber;
11
12  @override
13  Stream<Color> mapEventToState(ColorEvent event) async* {
14    |
15  }
16 }
17
```

Terminal output:

```
Built build/app/outputs/apk/debug/app-debug.apk.
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [3115681008] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [3115152624] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [2346214640] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread_join [3115681008] N3BRC18SingleThreadWorkerE succeed
F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread_join [3115152624] N3BRC18SingleThreadWorkerE succeed
F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread_join [2346214640] N3BRC18SingleThreadWorkerE succeed
```

Sekarang, mari kita hapus ini. Di sini, yang perlu kita lakukan hanyalah mengubah "\_color" menjadi warna baru berdasarkan event. Jadi, jika acara ini adalah "ColorEvent.to\_amber" maka yang akan saya kembalikan adalah "Colors.amber" lagi... biru muda. Jadi, jika "ColorEvent" adalah "to\_amber" maka "\_color" akan disetel ke kuning, jika tidak, akan disetel ke biru muda. Kemudian di sini, kami akan menghasilkan "\_color". "yield"



```
6 class ColorBloc extends Bloc<ColorEvent, Color> {
7   Color _color = Colors.amber;
8
9   @override
10  Color get initialState => Colors.amber;
11
12  @override
13  Stream<Color> mapEventToState(ColorEvent event) async* {
14    _color = (event == ColorEvent.to_amber) ? Colors.amber : Colors.lightBlue;
15    yield _color;
16  }
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

built build\app\outputs\apk\debug\app-debug.apk.

F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115681008] N3BRC18SingleThreadWorkerE

F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115152624] N3BRC18SingleThreadWorkerE

F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [2346214640] N3BRC18SingleThreadWorkerE

F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread\_join [3115681008] N3BRC18SingleThreadWorkerE succeed

F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread\_join [3115152624] N3BRC18SingleThreadWorkerE succeed

F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread\_join [2346214640] N3BRC18SingleThreadWorkerE succeed

adalah perintah yang akan memasukkan data ke dalam 'stream'. Seperti yang kita ketahui, ini mengembalikan 'aliran' bukan stream. Nah, "yield" menambah nilai (ke aliran masukan). Sama seperti saya menggunakan kran di tutorial sebelumnya, jadi kita tambahkan air lalu masukan stream aliran datanya. Pada dasarnya, hanya itu yang kita butuhkan untuk "BLoC" seperti ini saja

, jadi mari kita kembali ke 'main.dart'. Di 'main.dart' kita harus mengimpor terlebih dahulu. "import flutter\_bloc".

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_bloc/flutter_bloc.dart';
3
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatelessWidget {
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10       home: MainPage(),
11     ); // MaterialApp
12   }
13 }
14
15 class MainPage extends StatelessWidget {
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115081008] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115152624] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [2346214640] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread\_join [3115681008] N3BRC18SingleThreadWorkerE succeed  
F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread\_join [3115152624] N3BRC18SingleThreadWorkerE succeed  
F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread\_join [2346214640] N3BRC18SingleThreadWorkerE succeed  
Reloaded 0 of 462 libraries in 466ms.

Maka "MainPage" ini perlu dibungkus di dalam "BlocProvider". Di sini, kita akan menambahkan tipe - "ColorBloc" - "import"

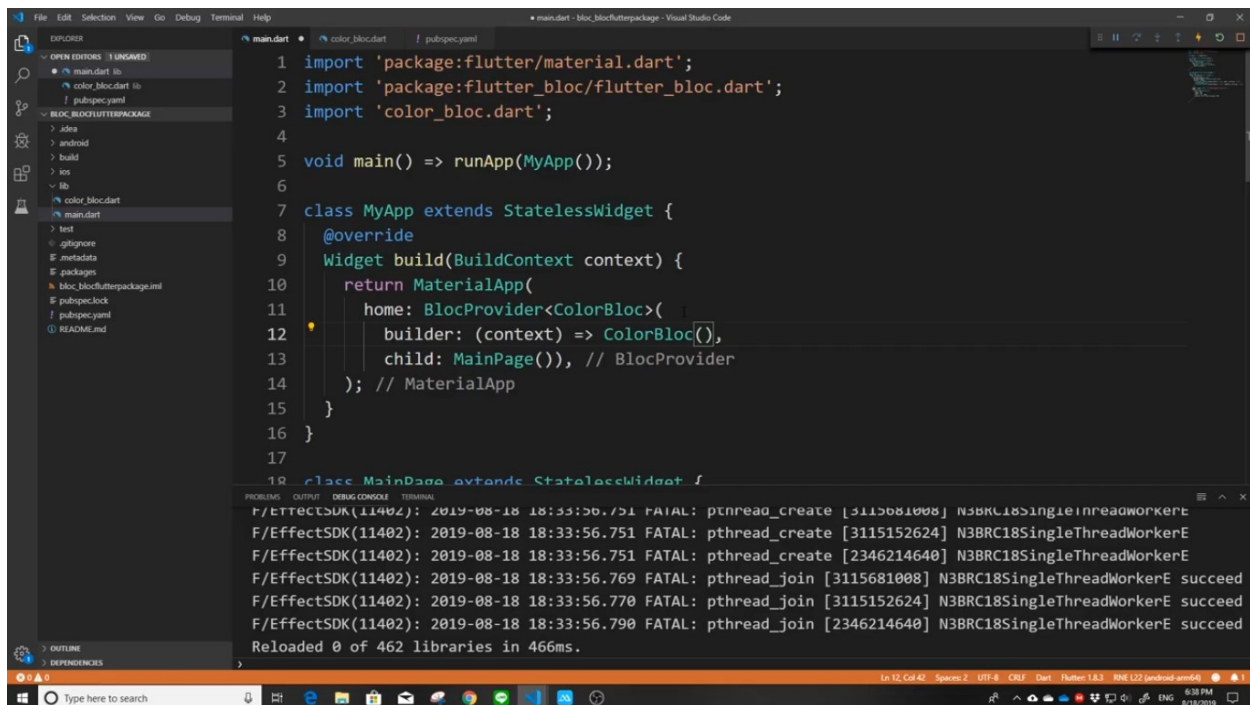
```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_bloc/flutter_bloc.dart';
3
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatelessWidget {
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10      home: BlocProvider<ColorBloc>{
11        // BlocProvider
12
13      }
14    );
15  }
16 }
17
18 class MainPage extends StatelessWidget {
19   @override
20   Widget build(BuildContext context) {
21     return Scaffold(
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115081008] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115152624] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [2346214640] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread\_join [3115681008] N3BRC18SingleThreadWorkerE succeed  
F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread\_join [3115152624] N3BRC18SingleThreadWorkerE succeed  
F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread\_join [2346214640] N3BRC18SingleThreadWorkerE succeed  
Reloaded 0 of 462 libraries in 466ms.

lalu di sini, tambahkan 'builder' "context..."

dan tambahkan metode yang akan mengembalikan "ColorBloc".

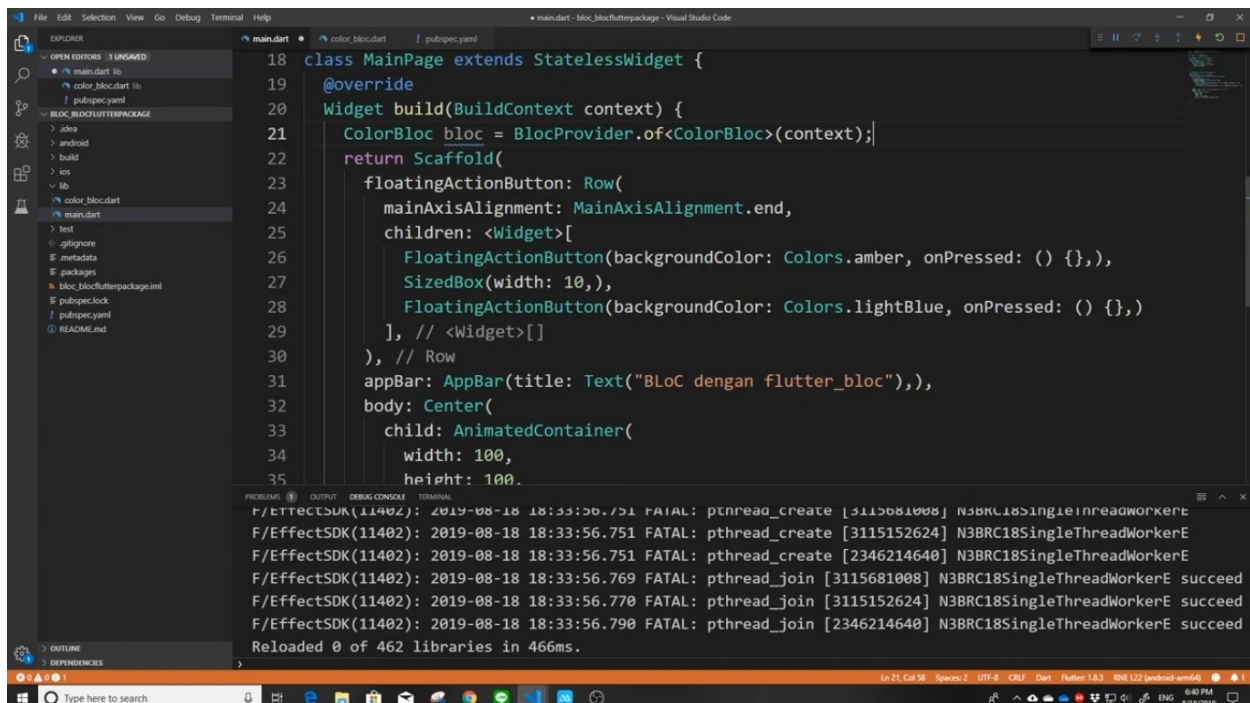


```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_bloc/flutter_bloc.dart';
3 import 'color_bloc.dart';
4
5 void main() => runApp(MyApp());
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: BlocProvider<ColorBloc>(
12         builder: (context) => ColorBloc(),
13         child: MainPage(), // BlocProvider
14       ); // MaterialApp
15   }
16 }
17
18 class MainPage extends StatelessWidget {
```

Terminal output:

```
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [3115681008] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [3115152624] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [2346214640] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread_join [3115681008] N3BRC18SingleThreadWorkerE succeed
F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread_join [3115152624] N3BRC18SingleThreadWorkerE succeed
F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread_join [2346214640] N3BRC18SingleThreadWorkerE succeed
Reloaded 0 of 462 libraries in 466ms.
```

Dengan ini, "ColorBloc" dapat diakses oleh semua 'anak' dari "BlocProvider" yang dapat ditemukan di "MainPage". Di "MainPage" di sini, di dalam metode "build" kami akan menambahkan "ColorBloc". "Bloc" sama dengan... "BlocProvider..." "dot of context" dan kita akan membuatnya bertipe "ColorBloc" Oke? titik koma.



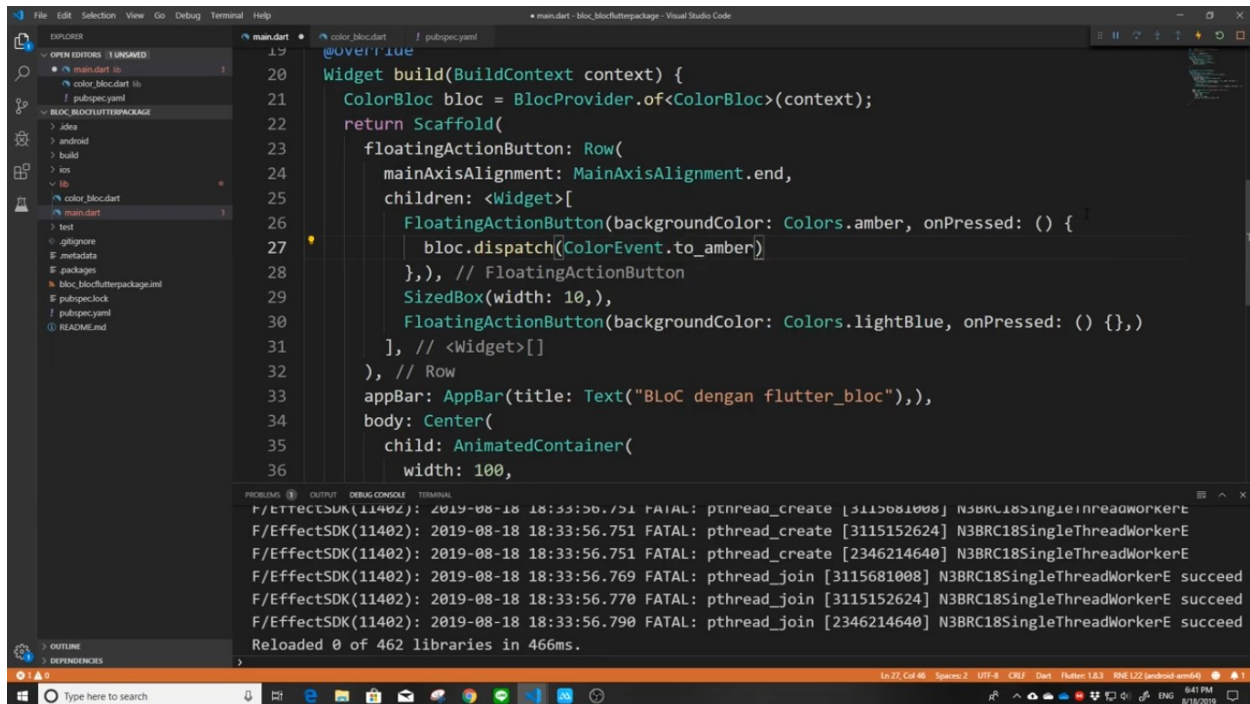
```
18 class MainPage extends StatelessWidget {
19   @override
20   Widget build(BuildContext context) {
21     ColorBloc bloc = BlocProvider.of<ColorBloc>(context);
22     return Scaffold(
23       floatingActionButton: Row(
24         mainAxisAlignment: MainAxisAlignment.end,
25         children: <Widget>[
26           FloatingActionButton(backgroundColor: Colors.amber, onPressed: () {}),
27           SizedBox(width: 10,),
28           FloatingActionButton(backgroundColor: Colors.lightBlue, onPressed: () {}),
29         ], // <Widget>[]
30       ), // Row
31       appBar: AppBar(title: Text("BLoC dengan flutter_bloc")),
32       body: Center(
33         child: AnimatedContainer(
34           width: 100,
35           height: 100,
```

Terminal output:

```
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [3115681008] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [3115152624] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread_create [2346214640] N3BRC18SingleThreadWorkerE
F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread_join [3115681008] N3BRC18SingleThreadWorkerE succeed
F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread_join [3115152624] N3BRC18SingleThreadWorkerE succeed
F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread_join [2346214640] N3BRC18SingleThreadWorkerE succeed
Reloaded 0 of 462 libraries in 466ms.
```

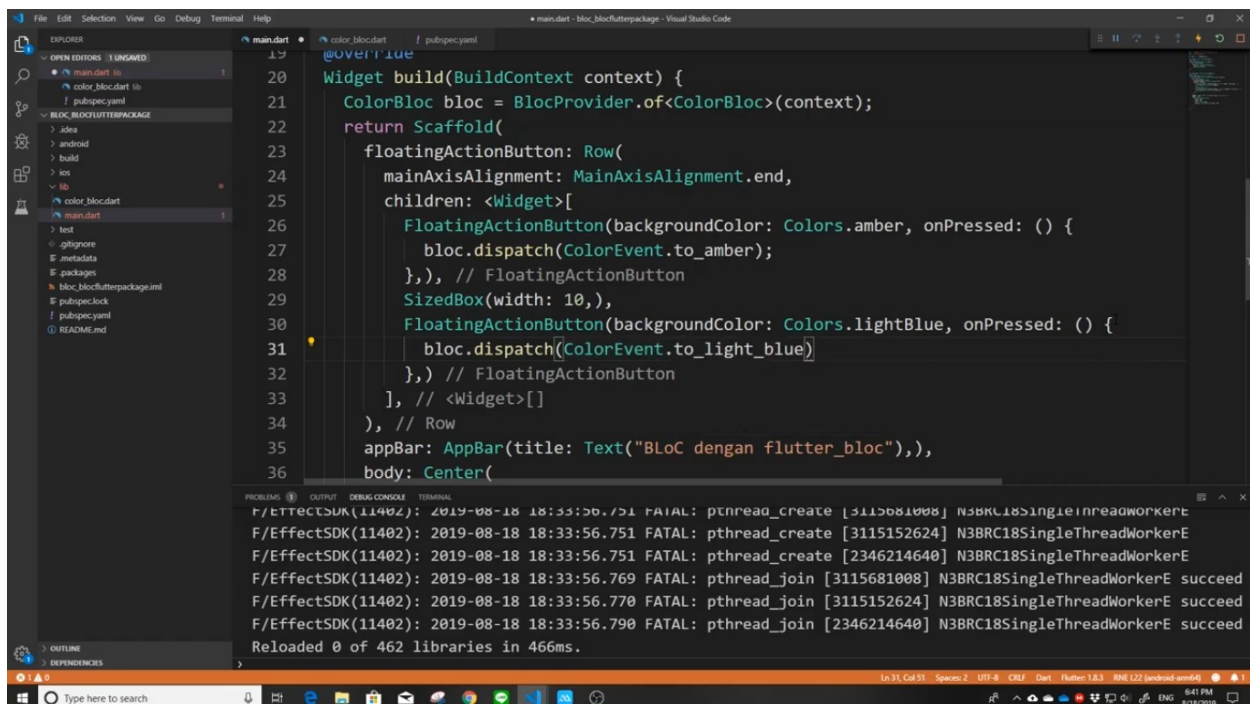
Ini adalah kode yang diperlukan untuk mendapatkan objek "blok" dari root "BlocProvider" - root di sini. Jadi, ini adalah objek yang akan kita ambil. Jadi di sini, ketika "FloatingActionButton" diklik, yang perlu kita lakukan hanyalah

menambahkan kode "bloc.dispatch..." diikuti dengan event yaitu "ColorEvent.to\_amber"



```
20 Widget build(BuildContext context) {
21   ColorBloc bloc = BlocProvider.of<ColorBloc>(context);
22   return Scaffold(
23     floatingActionButton: Row(
24       mainAxisAlignment: MainAxisAlignment.end,
25       children: <Widget>[
26         FloatingActionButton(backgroundColor: Colors.amber, onPressed: () {
27           bloc.dispatch(ColorEvent.to_amber);
28         },), // FloatingActionButton
29         SizedBox(width: 10,),
30         FloatingActionButton(backgroundColor: Colors.lightBlue, onPressed: () {}),
31       ], // <Widget>[]
32     ), // Row
33     appBar: AppBar(title: Text("BLoC dengan flutter_bloc")),
34     body: Center(
35       child: AnimatedContainer(
36         width: 100,
```

. Lalu, sama di sini, "bloc.dispatch..." "ColorEvent.to\_light\_blue". "Shift-Alt-F" dan simpan OK?

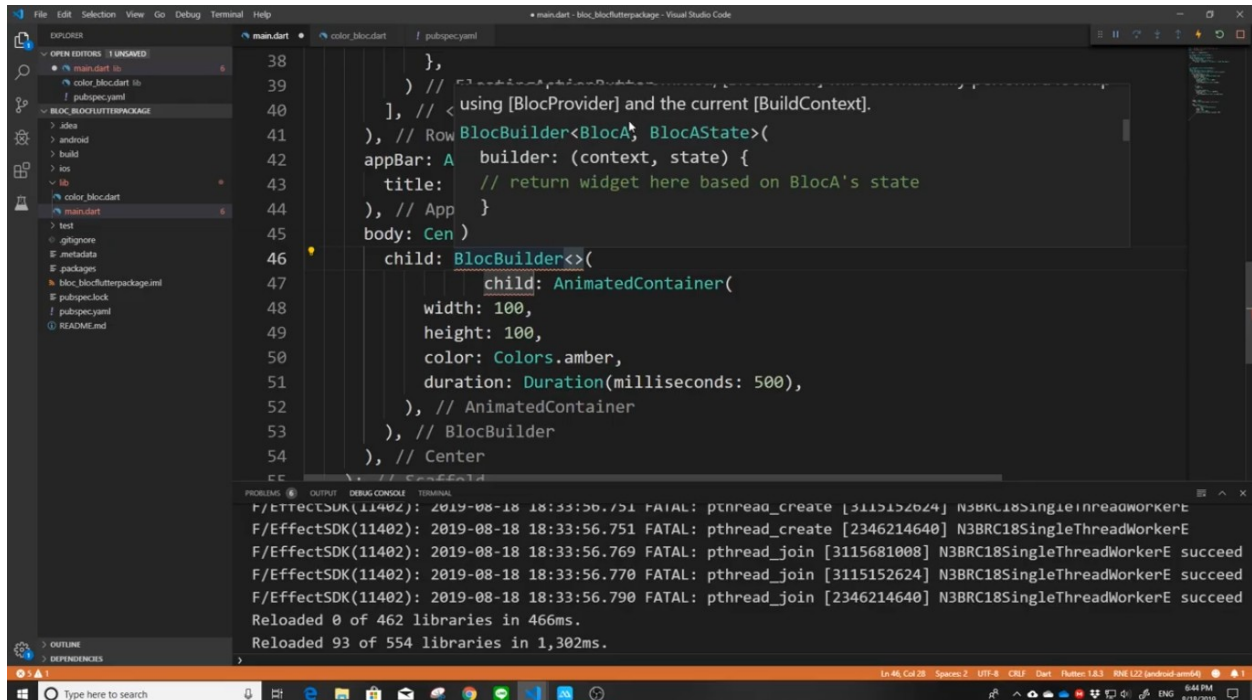


```
20 Widget build(BuildContext context) {
21   ColorBloc bloc = BlocProvider.of<ColorBloc>(context);
22   return Scaffold(
23     floatingActionButton: Row(
24       mainAxisAlignment: MainAxisAlignment.end,
25       children: <Widget>[
26         FloatingActionButton(backgroundColor: Colors.amber, onPressed: () {
27           bloc.dispatch(ColorEvent.to_amber);
28         },), // FloatingActionButton
29         SizedBox(width: 10,),
30         FloatingActionButton(backgroundColor: Colors.lightBlue, onPressed: () {
31           bloc.dispatch(ColorEvent.to_light_blue);
32         },), // FloatingActionButton
33       ], // <Widget>[]
34     ), // Row
35     appBar: AppBar(title: Text("BLoC dengan flutter_bloc")),
36     body: Center(
```

Mari kita periksa. Saya lupa menjelaskan bahwa "BlocProvider" juga bertugas menangani metode pembuangan, setelah "BLoC" tidak digunakan lagi. Jika Anda ingat, di video sebelumnya kami membuat metode "pembuangan" kami sendiri,



tetapi sekarang tidak perlu karena semuanya ditangani oleh "BlocProvider". Oleh karena itu, juga tidak perlu menggunakan widget 'Stateful', widget 'Stateless' sudah cukup. Mari kita gulir ke bawah. Yang tersisa untuk dilakukan sekarang adalah membungkus 'AnimatedContainer' dengan 'BlocBuilder'. Kita perlu menambahkan tipe untuk "BlocBuilder", mari kita lihat... Ada contohnya di sini.

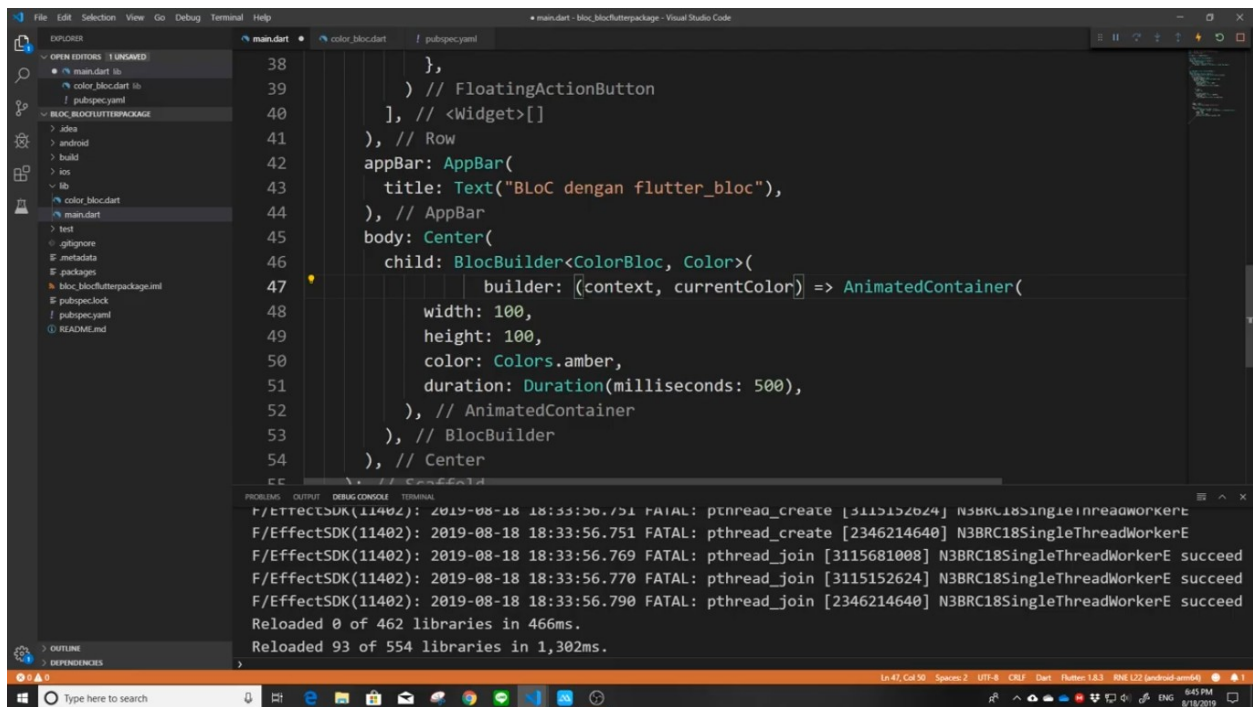


```
38      },
39      ) // <using [BlocProvider] and the current [BuildContext].
40    ], // <
41  ), // Row
42  appBar: A builder: (context, state) {
43    title: // return widget here based on BlocA's state
44  }, // App
45  body: C()
46  child: BlocBuilder<BlocA, BlocAState>(
47    child: AnimatedContainer(
48      width: 100,
49      height: 100,
50      color: Colors.amber,
51      duration: Duration(milliseconds: 500),
52    ), // AnimatedContainer
53  ), // BlocBuilder
54  ), // Center
55  ), // Center
```

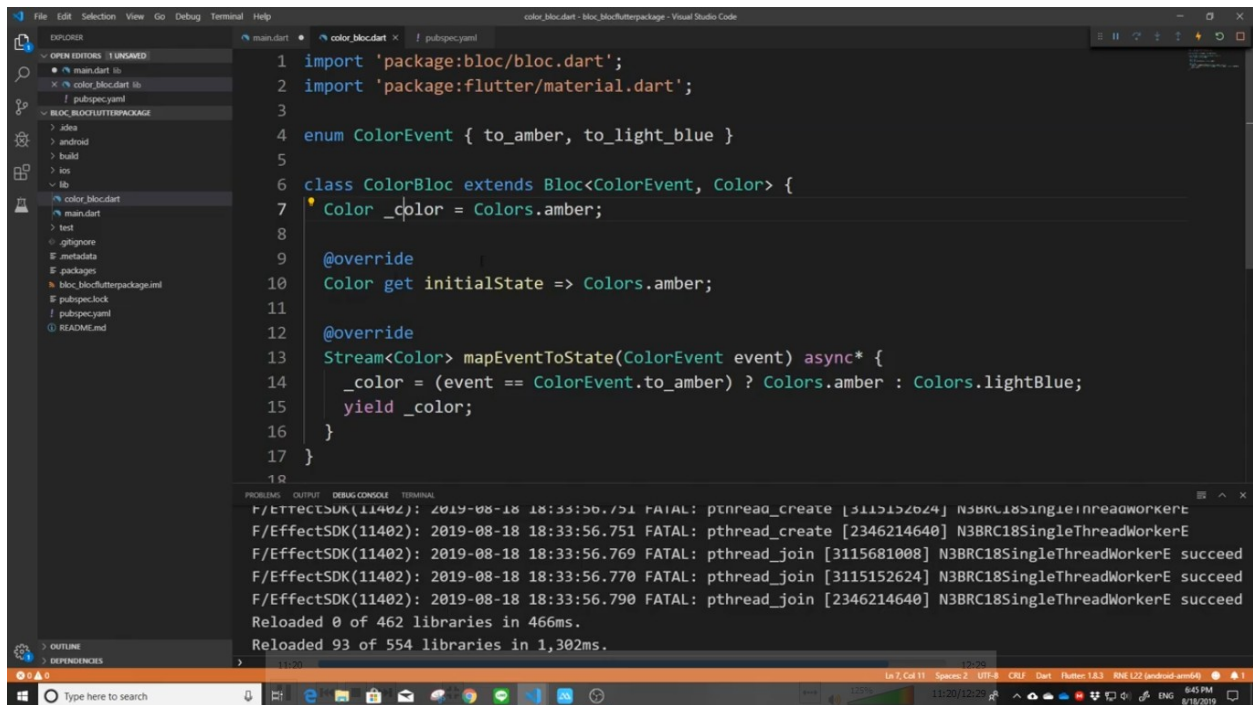
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [3115152624] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.751 FATAL: pthread\_create [2346214640] N3BRC18SingleThreadWorkerE  
F/EffectSDK(11402): 2019-08-18 18:33:56.769 FATAL: pthread\_join [3115681008] N3BRC18SingleThreadWorkerE succeed  
F/EffectSDK(11402): 2019-08-18 18:33:56.770 FATAL: pthread\_join [3115152624] N3BRC18SingleThreadWorkerE succeed  
F/EffectSDK(11402): 2019-08-18 18:33:56.790 FATAL: pthread\_join [2346214640] N3BRC18SingleThreadWorkerE succeed  
Reloaded 0 of 462 libraries in 466ms.  
Reloaded 93 of 554 libraries in 1,302ms.

Yang pertama adalah jenis "Blok" dan yang kedua adalah "state". Jadi, di depan kita akan menambahkan "ColorBloc" sedangkan di belakang, kita akan menambahkan "Color". "ColorBloc" koma "Color" lalu tambahkan 'builder' di sini. Ini akan membutuhkan metode yang akan mengembalikan widget yang akan dibuatnya yaitu 'AnimateContainer'. Metode itu sendiri memerlukan dua parameter, yang pertama adalah 'konteks', dan yang kedua adalah color yang merupakan warna saat ini. Jadi, jika ada perubahan 'status', 'warna' di "BLoC" kami akan mengirimkan warna saat ini. Jadi di sini kita menulis "konteks" koma...currret color "Warna saat ini"

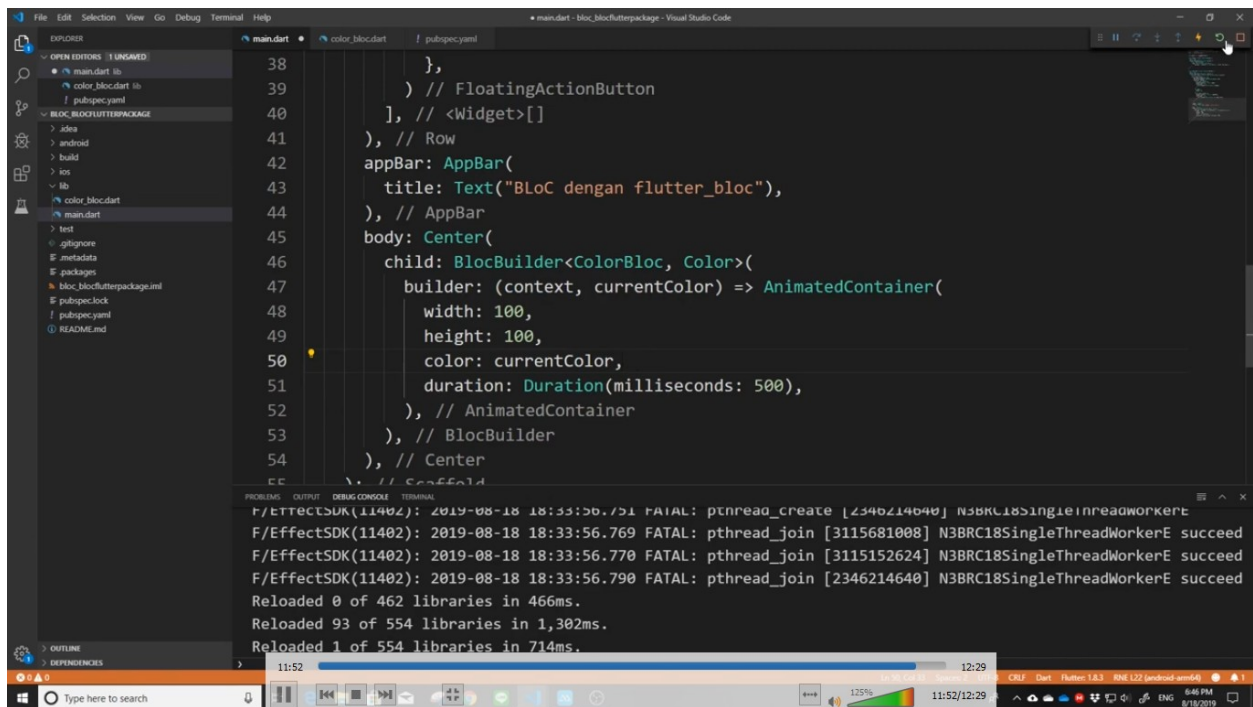


. Jadi, warna saat ini akan diganti dengan ini.

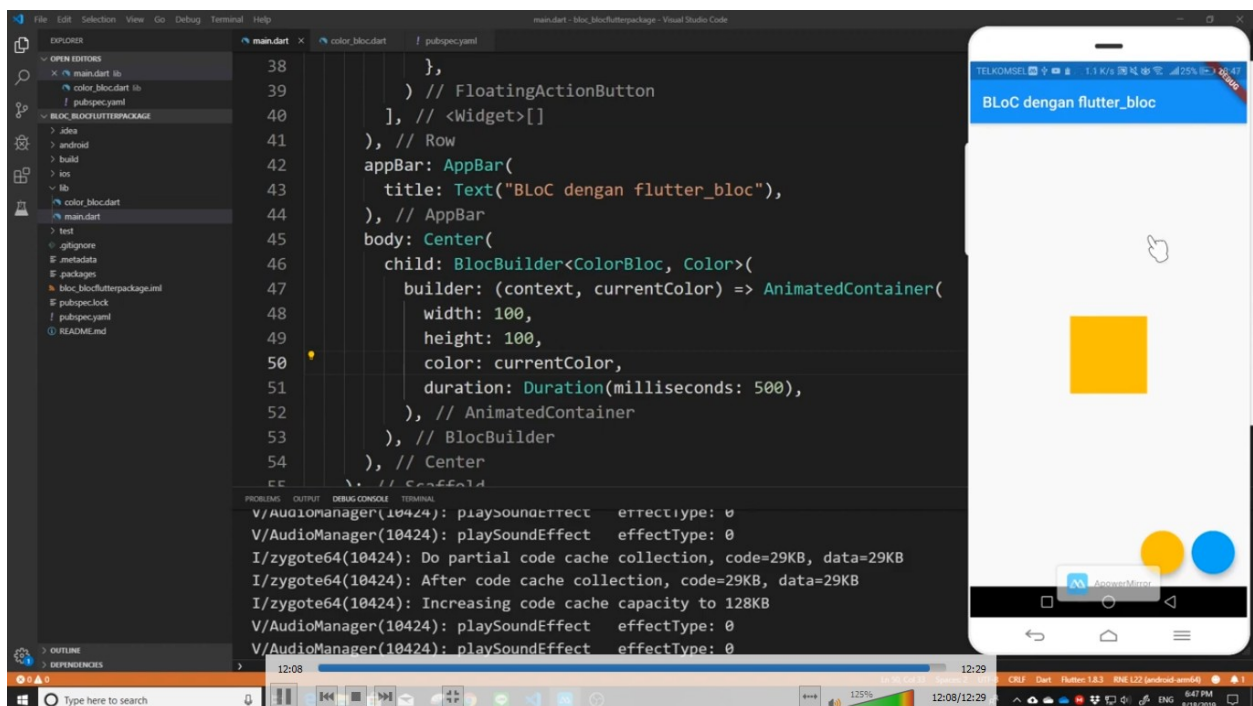


"\_color", warna yang ada di "ColorBloc" akan diganti dengan warna terbaru. Ini akan diubah kan? Dan setelah diubah, itu akan dimasukkan ke dalam stream. Setelah dimasukkan ke stream, itu akan ditangkap di sini. Jadi "currentColor" akan menyimpan ini, warna baru. Jadi yang perlu kita lakukan hanyalah mengubahnya dengan "currentColor". Menyimpan. "Shift-Alt-F".





Sekarang, mari kita restart. Mari kita mengujinya. Berhasil! Itu berubah.



OK, jadi begitulah cara Anda melakukannya. Bagaimana Anda membuat "BLoC State Management" menggunakan paket "flutter\_bloc". Sangat sederhana bukan? Saya harap, dengan tutorial video ini Anda dapat, dengan mudah, mengimplementasikan "BLoC State Management" dalam aplikasi Android Anda. Terima kasih telah menonton, sampai jumpa di video saya yang akan datang.